

```

SELECT  student_id,
        student_name,
        grade
FROM    students AS A
INNER JOIN grades AS B
ON A.student_id = B.student_id

```

Student_id	Student_name	grade
2	Bob	B
3	Charlie	A

Q2 - Left Join

```

SELECT  emp_id,
        emp_name,
        dept_name
FROM    employees AS A
LEFT JOIN departments AS B
ON A.emp_id = B.emp_id ;

```

emp_id	emp_name	dept_name
1	John	null
2	Lisa	HR
3	Mike	null

Q3 - FULL OUTER JOIN

```

SELECT  product_id,
        product_name,
        quantity
FROM    products AS A

```

FULL OUTER JOIN sales AS B
ON A.product-id = B.product-id;

product-id	product-name	quantity
1	laptop	null
2	mouse	50
3	keyboard	null
4	null	30

Q4 - LEFT JOIN + CASE

SELECT orders_id,
customer_id,
amount,
customer_name

FROM orders AS A

LEFT JOIN customers AS B

ON A.customer-id = B.customer-id;

order_id	customer_id	amount	customer_name
101	101	500	Paul
102	102	300	Sarah
3	105	0	null

```

5. SELECT A.region-id,
        region-name,
        SUM (amount) AS total-sales
FROM regions AS A
LEFT JOIN sales AS B
ON A.region-id = B.region-id
GROUP BY A.region-id, region-name;

```

region-id	region-name	total-sales
1	North	2 000
2	South	2 500
3	East	NULL

```

6. SELECT A.student-id,
        name,
        days-present,
        CASE
            WHEN days-present >= 15 THEN 'Excellent'
            WHEN days-present BETWEEN 6 AND 14 THEN 'Needs Improvement'
            WHEN days-present <= 5 THEN 'Poor Attendance'
            ELSE 'No Record'
        END AS attendance-status
FROM students AS A
LEFT JOIN attendance AS B
ON A.student-id = B.student-id;

```

student-id	name	days-present	attendance-status
1	Alice	15	Excellent
2	Bob	5	Poor Attendance
3	Charlie	Null	No Record


```

7. SELECT A.project_id,
           name,
           COUNT(task) AS task_count
FROM projects AS A
INNER JOIN tasks AS B
ON A.project_id = B.project_id
GROUP BY A.project_id, name;

```

project-id	name	task-count
1	AI Chatbot	2
2	Website	1

```

8. SELECT COALESCE (A.cust_id, B.cust_id) AS cust_id,
           order_total,
           return_total,
           CASE
             WHEN return_total IS NOT NULL THEN 'Returned'
             ELSE 'No Return'
           END AS return_status
FROM orders AS A
FULL OUTER JOIN returns AS B
ON A.cust_id = B.cust_id
WHERE order_total > 100

```

cust-id	order-total	return-total	return-status
11	120	20	Returned
12	250	Null	No Return
13	180	Null	No Return

```

9. SELECT A.user_id,
        name,
        COUNT (login_date) AS login-count
FROM users AS A
LEFT JOIN logins AS B
ON A.user_id = B.user_id
GROUP BY A.user_id, name
ORDER BY login-count, DESC;

```

user_id	name	login-count
2	Gloria	2
3	Steve	1
1	Nelson	0

```

10. SELECT A.teacher_id,
        teacher_name,
        COALESCE (subject_name, 'No Subject Assigned') AS subject_name
FROM teachers AS A
LEFT JOIN subjects AS B
ON A.teacher_id = B.teacher_id
ORDER BY teacher_name ASC;

```

teacher_id	teacher_name	subject_name
3	Mr Dlamini	No Subject Assigned
1	Mr Hlongwane	Math
1	Mr Hlongwane	Science
2	Ms Ndaba	No Subject Assigned